

Cloud Connectivity and Embedded Sensory Systems

Lambert Spaanenburg · Hendrik Spaanenburg

Cloud Connectivity and Embedded Sensory Systems

 Springer

Lambert Spaanenburg
Department of Electrical & Information
Technology
Lund University
S-221 00, Lund, Sweden
lambert@eit.lth.se

Hendrik Spaanenburg
Heterogeneous Computing, LLC
Durham, NH 03824, USA
henk@heterogeneouscomputing.com

ISBN 978-1-4419-7544-7

e-ISBN 978-1-4419-7545-4

DOI 10.1007/978-1-4419-7545-4

Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2010937673

© Springer Science+Business Media, LLC 2011

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

To Henk Sr.

Preface

Computers, great or small, can be found in many digital data producing and processing system nodes. Their performance is steadily increasing, and some have already become miniature supercomputing nodes. More and more of them are connected to the Internet, thereby physically forming very large networks.

The classical networks contain computers that are cabled together. History shows two developments here. On the one hand, we see mainframes becoming personal computers and then notebooks, getting more personal all the time. On the other hand, we see mainframes turning into supercomputers and then server farms into the cloud, getting more into a shared service all the time.

There are many stories about the origin of “cloud computing.” The most popular one is that people were diagramming an information-processing architecture on a whiteboard. In order to sketch out the magical process that was finally to do everything, some wrinkles were drawn in the sky and they called it a cloud. This is not a rigid definition and consequently the name “cloud” is used for a lot of things.

A cloud provides shade against the sun. To be in the shade, you have to move regularly. If you do not want that, then take an umbrella. It does not give shade to a lot of people, but it will help you. More or less the same is true for the rain cloud. You can walk around with a pot to water your plants, but it is just a bit of water at a time. Having a cloud makes it much easier.

In the industrial age, we have become accustomed to mass fabrication, mass transport, and mass delivery. Already the Romans had structures to gather and transport water, in order to make it massively available in the city. Otherwise you need to live near the well, and when the well goes dry you have to move. Later the same principle was applied for electricity, again with the main benefit that massive use becomes possible without the need to live next to the generator. In the information age, we want essentially the same, but this time for information and specifically sensor-obtained information.

The Internet has become the aqueduct of modern times, bringing information from the large server farms to each household. At the origin, it is not really a cloud, but let us not be too picky. The point we want to make here is to take the cloud as part of a shared resource in a cyclic bio-like system. This would bring us closer to

the concept of “sensory networks.” In talking about such networks, we can easily draw up the network nodes, but to indicate the overall processing function some wrinkles in the sky have to be drawn. Just like a cloud!

The point is that a function can be part of a node, can use some nodes, and can also be handled at a server farm. The function is executed on some resources and we do not want to specify immediately where these resources are located and how much of them are needed. It should be moving dependent on the need, and we do not want to know that and where it moved. In other words, *the network becomes the cloud!*

The nature of the network has also changed, though the backbone is still cabled. However, the introduction of wireless technology has caused the computer to become mobile. And from there, we see other computerized mobile equipment coming into fashion, notable the mobile telephone. Yearly 1.5 billion camera phones are sold!

The rich set of user interactions is currently being augmented with inter-device interactions, making the camera phone a flexible node in a community network. Being based on preferential attachments, its function and construction changes with the whim of the participants rather than on pre-composed orchestration.

But the story does not stop with phones. All kind of products are turning to wireless, if not only to get rid of the cables. The doorbell connects wirelessly to the phone and allows you to answer the doorbell, even when you are not at home. The car seat for the children warns for dehydration of the baby. And in all these new devices, we have computing platforms ranging from simple controllers to graphic accelerators.

Similar networking trends such as in wireless consumer communities are visible in smart energy grids, homeland security systems, and environmental networks. These sensory networks are meant to create awareness in space and time. They may be measuring the presence of an object or a condition, characterizing an object stream or a situational pattern, or even detect abnormalities that are to occur. Different types of sensors can be used for similar purposes, allowing for enhanced system reliability and safety, but they tend to be preferentially attached to the network.

For the network functionality, one can distinguish between cloud, and even swarm and flock computing, the difference being the distribution of the decision control. Just plugging sensors into the cloud is one thing, but the question remains how the extended network itself can be made more intelligent and therefore less dependent on the quality of the individual sensors. We call this *embedded sensory systems*.

Sensory systems are populated by embedded functions that safely combine migrating software on virtualized hardware. It brings the connectivity mechanisms from the global cloud into local clouds of sensor-activated resources. This emphasizes safety for society through ownership of data by a rigorous awareness for the users and their intentions. Such systems will leave a clear mark on society in the coming years.

The Book

It provides conceptually a one-stop entry into the world of intelligent sensory networks, as we envision them to exist in relationship to the computing cloud(s). This is the “ambient intelligence” equivalent of what “cloud computing” means to business intelligence, and therefore the new generation of large-scale sensory networking systems. Cloud computing usually refers to the accessibility from the network of a well-orchestrated data warehouse. Clearly, cloud computing concepts also apply to networks of embedded sensors. The book discusses concepts, problems, and initial solutions for the latter, while also providing illustrations from a wide field of practice. Becoming aware of the basic problems and solutions helps the readers to prepare for innovation based on distributed intelligence in their own area of technological expertise.

This book provides new theory on the design of wireless sensory networks. It provides a step-by-step discourse on building case studies to capture the requirements, taking into account practical limitations to creating ambient intelligence. Notably, it pays attention to topological and communication constraints, while adding intelligence, security, and safety to the overall desired system functionality. Such aspects will be highlighted by reviewing applications in a variety of typical sensory/cloud network scenarios.

The reader will not only achieve a better understanding of such concepts as sensory clouds but will also be guided by examples on how to design such networks, taking the typical characteristics of diverse application areas into account.

In general, knowledge on new technologies is communicated in new conferences and journals, and in addition needs to be condensed regularly in books. This book will give the readers an overview of the current directions in theory and practice and will therefore allow them to stand on the shoulder of “giants”/peers. Our emphasis in writing this book has been on developing the new concepts, especially the disruptive technology aspects, for use in the establishment of new businesses.

The book is divided into four parts. The first part explores the opportunities that will come about from the further developments of cloud computing into the outlying domains of sensory systems. It also includes a description of a new software model for large-scale networks. The second part reviews system design aspects needed for the understanding of the subsequently illustrated potential sensory network applications. The emphasis is on system applications that would benefit from the connection into the Cloud. The third part addresses concerns for security and safety that will need to be supported in order to result in successful cloud-centric sensory networks. Lastly, we focus on details of a futuristic sensory/display-instrumented home environment completely in touch with the Cloud.

This book aims to be one of the first to bring the sensor-networking-into-the-cloud state-of-the-art together with a perspective on the near future to stimulate further research. It provides a selection of material from courses at Lund University (Sweden) and the Daniel Webster College in Nashua (New Hampshire, USA). It brings a vision that has gradually evolved from the experience with

a number of students. Especially our gratitude goes to Wen Hai Fang, Peter Sundström, Lars Lundqvist, Erik Ljung, Erik Simmons, Barbara Olmedo, Sajjad Haider, Joe Evans, Dalong Zhang, Miao Chen, Mona Akbarniai Tehrani, Shankar Gautam, Olle Jakobsson, Lijing Zhang, Wang Chao, Shkelqim Lahi, Nabil Abbas Hasoun, Dongdong Chen, Bintian Zhou, Cheng Wang, Deepak Gayanana, Markus Ringhofer, Vince Mohanna, Johan Ranefors, Suleyman Malki, Jos Nijhuis, and Ravi Narayanan.

Lund Sweden
Durham New Hampshire, USA

Lambert Spaanenburg
Hendrik Spaanenburg

Contents

Part I Clouds into the Sensors' Edge

1	Introduction to Cloud Connectivity	3
1.1	Embedded Systems	4
1.1.1	Characteristics of Embedded Systems	5
1.1.2	What Makes Embedded Systems Special	7
1.1.3	Encapsulated Architecture	8
1.1.4	Dichotomized Architecture	9
1.1.5	Expanded Architecture	9
1.2	Network Processing Architectures	10
1.2.1	Statically Structured Network Processing	10
1.2.2	Dynamically Structured Network Computing	12
1.2.3	The Network Is the Computer TM	15
1.3	Intelligent Sensor Networking	15
1.3.1	Various Sensing Principles	16
1.3.2	Sensory Networks	17
1.3.3	Sensor Networks Design Methodology	19
1.4	Expansion of the Ambient Intelligence Approach	21
1.4.1	Situational Computing	22
1.4.2	Autonomic Computing	23
1.4.3	Organic Computing	24
1.5	Cloud Computing Concepts	25
1.5.1	The Road to the Cloud	26
1.5.2	Commercial Clouds	28
1.5.3	Server Farms	30
1.5.4	Network Extenders	30
1.6	Cloud-Sourcing	32
1.6.1	Business Models	32
1.6.2	Cloud Economics	34
1.7	Internet-of-Things	35
1.7.1	Very-Thin-Clients	36
1.7.2	Android-in-the-Cloud	36

1.7.3	Sensing-into-the-Cloud	38
1.8	Summary	40
	References	41
2	Software-from-the-Cloud	43
2.1	The Nature of the Cloud	45
2.1.1	Virtualization	46
2.1.2	Looking into the Cloud	49
2.1.3	The Other Cloud	52
2.2	Connection-Set Architecture	53
2.2.1	Software Connects	53
2.2.2	Skeleton Injector	54
2.2.3	Synchronization	56
2.3	Software Migration Concept	57
2.3.1	Software Components	58
2.3.2	Introducing AUTOSAR	60
2.3.3	The AUTOSAR Case	62
2.4	Topological Impact	68
2.4.1	Homological Sensor Networks	69
2.4.2	Configuration of Embedded Software	71
2.4.3	Adaptive Reconfiguration	73
2.5	SoftWare Is Virtual HardWare!	74
2.5.1	Elasticity	75
2.5.2	Heterogeneity	78
2.5.3	Optimization	78
2.6	Summary	79
	References	79

Part II Cloud-Centric Systems

3	System Requirements, Understanding, and Design Environment	85
3.1	System Design Methodology	86
3.1.1	The Traditional Ways	86
3.1.2	Embedded System Design	89
3.1.3	Model-Based Design	91
3.2	Neural Network System Control	93
3.2.1	Neural Network Theory	94
3.2.2	Neural Control	97
3.2.3	Neural Controller Design	98
3.2.4	The Modular Hierarchy	102
3.3	Networked System Design	105
3.3.1	Making a Story	106
3.3.2	The Path Diagram	109
3.3.3	The Wiring Diagram	110
3.3.4	The Sequence Diagram	110

3.4	System Case Study	111
3.4.1	Case Outline	112
3.4.2	Refining the Story	114
3.4.3	Moving to Scenes	115
3.5	Summary	117
	References	117
4	Sensor-Centric System Developments	121
4.1	Wireless Sensor Network Technologies	122
4.1.1	Wireless Communication Protocols	123
4.1.2	Power Management	128
4.1.3	Power Harvesting	131
4.2	Personal-Area Networks	131
4.2.1	Body Sensor Networks	131
4.2.2	(Serious) Gaming	134
4.2.3	Business- and Edu-tainment	134
4.3	Monitoring and Observation	135
4.3.1	Wear-Free Transmission	135
4.3.2	Care, Comfort, and Concern Monitoring	136
4.3.3	Security Surveillance	137
4.3.4	Environmental Monitoring	139
4.4	Monitoring and Control	140
4.4.1	Smart Structures	140
4.4.2	Traffic Control	141
4.4.3	Smart Power Grid	142
4.4.4	Industrial Automation	143
4.5	Collective Intelligence	143
4.5.1	Collision Avoidance	146
4.5.2	Tracking in a Billiard System	149
4.5.3	Trajectory Modeling	151
4.5.4	Cellular Neural Network Learning	157
4.5.5	Moving Target Detection	158
4.6	Multi-sensor Intelligence	167
4.6.1	Intelligence Gathering	167
4.6.2	Multi-sensor Fusion	168
4.7	Summary	169
	References	170

Part III Everything-in-the-Cloud

5	Security Considerations	177
5.1	Reliability	177
5.1.1	Basic Concepts	178
5.1.2	Probing Cases	179
5.1.3	Software Metrics	181
5.2	Trustability	182

5.2.1	Trusted Circuits	182
5.2.2	Trusting-in-the-Edge	185
5.2.3	Secure Mobility	186
5.3	Resiliency	187
5.3.1	Fault Tolerance	187
5.3.2	Fault-Tolerance in Computing	189
5.3.3	Secure Communications	190
5.3.4	Redundancies in Large-Scale Sensor Networks	191
5.4	Authentication	195
5.4.1	Sensor Accreditation	196
5.4.2	Sympathetic Testing for Trustability Assessment	196
5.4.3	Remote Entrusting	199
5.5	Security-as-a-Service	199
5.6	Summary	201
	References	201
6	Safety Considerations	205
6.1	Adversity Action Detection	205
6.1.1	Jamming	207
6.1.2	Tampering	207
6.1.3	Invading	207
6.1.4	Antipathetic Testing for Malfeasance Assessment	207
6.2	Byzantine Generals	208
6.2.1	Faults	208
6.2.2	Signed Messages	209
6.2.3	Missing Communications	209
6.3	Emergent Behavior Detection	209
6.3.1	Information Reporting and Management	209
6.3.2	Anomaly Detection	212
6.3.3	Nuisance Alarms	214
6.3.4	Redundancy by Intelligent Agents	216
6.4	Identity Assurance	218
6.4.1	Biometric Authentication	218
6.4.2	BASE for Multi-modal Sensing	222
6.5	Safety in Transactions	224
6.5.1	e-Traveling Card	225
6.5.2	The i-Coin	232
6.6	Safety-as-a-Service	234
6.7	Summary	234
	References	234
Part IV	Last-Mile-from-the-Cloud	
7	Bringing the Cloud Back to Home	239
7.1	Net-Centric Home Automation	239
7.1.1	Home-Networks anno 2000	240

7.1.2	Operating the Home anno 2010	243
7.1.3	The Home Outside the House	244
7.2	Home–Human Interfaces	246
7.2.1	Home Imagers	246
7.2.2	Panoramic Visualizations	252
7.3	Media Inviter	260
7.3.1	Integrated Media Community	260
7.3.2	Home-Inviter Interface	264
7.3.3	Multimedia Home Automation	267
7.3.4	Multimedia Home Security	268
7.4	Futuristic Home Environment	269
7.4.1	The Intelligent Ball	269
7.4.2	Complexities of Operation	270
7.4.3	HPC-from-the-Cloud	271
7.4.4	Green Computing	275
7.5	Summary	275
	References	276
8	Afterwords	279
8.1	Cloud	279
8.2	Sensor	280
8.3	And Everything in Between	281
8.4	Panta Rei	282
Index		283

List of Figures

1.1	Embedded systems will be everywhere, but mostly unnoticeable or invisible, to enhance the functionality of the devices and equipment shown	5
1.2	Embedded systems in their environment	6
1.3	A computing node in a dichotomized architecture	9
1.4	The influx of sensory networks [14]	17
1.5	Current bottom-up network design	19
1.6	Future top-down network design	20
1.7	Bouncing against the barriers	26
1.8	Femtocell becoming mainstream within 3G	31
1.9	Datacenter is the computer	35
1.10	Android architecture [37]	37
1.11	Integration of “cloud computing” and “ambient intelligence”	38
1.12	Dualism in polytopol computing between multi-core processors and distributed sensor systems	39
2.1	From (a) Mainframe over (b) Client/server to (c) The embedded cloud	44
2.2	Virtualization concept development timeline	46
2.3	The coming of age of the hypervisor	47
2.4	(a) Monolithic and (b) Console-Guest hypervisor [29]	48
2.5	(a) Monolithic and (b) Native hypervisor [28]	49
2.6	Workload (a) Isolation, (b) Consolidation, and (c) Migration	50
2.7	The augmented phone	51
2.8	Abstraction layers built through virtualization [29]	51
2.9	Java versus skeleton injector	55
2.10	Scheduling versus injection	55
2.11	Basic interaction architecture	57
2.12	AUTOSAR technical scope	61
2.13	The AUTOSAR ECU software architecture	63
2.14	BSW Components	64
2.15	Types of SWCs	65
2.16	Overview of SWC with different interfaces	66
2.17	Description for internal and external behavior	66

2.18	Relation of AUTOSAR components to header files	68
2.19	Shadow changes with light	70
2.20	Homological sensor networks [30]	71
2.21	Configuration phases	71
2.22	Transparent BSW/SWC communication architecture	73
2.23	Game of life examples	74
2.24	Makimoto's wave extended	75
3.1	The waterfall approach	87
3.2	Böhm's spiral development [2]	87
3.3	Computing and communication technologies evolution: 1960–2010! [3]	89
3.4	The overall design flow for embedded system design	90
3.5	A walk through the system	91
3.6	The sequence diagram	92
3.7	Meal manager state diagram	92
3.8	The operation of a single neuron	95
3.9	A multilayer feed-forward network	95
3.10	Direct (a) and Indirect (b) Neural control	97
3.11	Hierarchical alarm handling in rudder control [16]	98
3.12	The general human driver/car control system (a), the neural network/car control system (b), and a human driving behavior identification model	99
3.13	The steering behavior of the human driver compared with the neurocontroller	101
3.14	The three-tier concept	103
3.15	Plot of target values for the lower networks (The training data represents the whole problem space.)	105
3.16	Opposite level traversing in design and exploration [31]	106
3.17	Design path in UML	108
3.18	Layer meaning	108
3.19	Path diagram for the virtual food preparation	109
3.20	Wiring diagram for the virtual food preparation	110
3.21	Conveyor belt sequence diagram [35]	111
3.22	Path diagram	114
3.23	A surveillance hierarchy	115
3.24	Wiring diagram	116
3.25	Mike is talking to the iCAT	117
4.1	Range and data rate tradeoffs in protocols (After Freescale Semiconductor)	124
4.2	TinyOS components [9]	126
4.3	Sample TinyOS application framework [9]	127
4.4	ZigBee stack architecture [10]	128
4.5	Bluetooth	129
4.6	A layered power management controller	130

4.7	Example power densities of energy harvesting mechanisms [16]	131
4.8	Body-area watt-level available [17]	132
4.9	MIT sociometrics	135
4.10	Hierarchical layering of awareness	137
4.11	Schematic view of a LOFAR node [72]	140
4.12	Concept of operations, quick highway incident detection and incident warning system [26]	141
4.13	Autonomous intelligent vehicle [27]	142
4.14	NXP Semiconductors' WiCa wireless smart camera board	145
4.15	WiCa block diagram	146
4.16	Basic driving situations	147
4.17	(a) Screen showing a driving environment with wind (<i>arrows</i>) and ice (<i>shaded roads</i>) and (b) Screen showing the actual position, direction, and range of the thirteen distance sensors on the car	148
4.18	Sinai billiard ball simulator	149
4.19	Different motion patterns of the ball	150
4.20	Main framework in event detection task	152
4.21	Separating FG from BG pixels by hole filling	152
4.22	Top shows Temporal Difference Method (<i>left</i>) and Background Updating Method with Learning Rate 0.5 (<i>right</i>). Bottom shows Filled Image with the Method described here (<i>left</i>), and the Final Result after finding the Blob Borders (<i>right</i>)	154
4.23	Trajectories in a scene with more than one path	155
4.24	A set of observed walking trajectories	156
4.25	Dataflow in learning phase	156
4.26	Dataflow of testing phase	157
4.27	Block diagram of a DT-CNN cell	158
4.28	Dimensionality of CNN image processing	159
4.29	Mapping of the image on the pixel map	160
4.30	Pixel displacement versus observation distance for several object velocities	161
4.31	Flow diagram for the template application	162
4.32	Measuring the displacement of an object moving from right to left in the scenery. Displacement (shown in (c)) of the moving object is the difference BETWEEN the black boxes in (a) and (b)	162
4.33	First two frames (f_1 and f_2) of the video sequence after applying the averaging template for a number of iterations	163
4.34	(a) Resulting image of the absolute value of the difference of two consecutive frames. Darkest pixels are observed where the two frames differ as most. (b) Intermediate result	

	after skeletonization, where the isolated pixels can easily be noticed	164
4.35	Applying the template of IPR removes all isolated pixels (a). Procedure of segmentation is completed once the binary mask is created (b)	164
4.36	The intermediate results of all steps as obtained from the post place and route simulation	165
4.37	Separation between blobs due to different speeds: “Slow” object in (a) and a “Fast” one in (b). The arrows indicate the direction of the movement	166
4.38	Extended algorithm for handling fast-moving objects. the direction of movement is from <i>right</i> to <i>left</i>	167
5.1	Product development life-cycle	180
5.2	Controlled and uncontrolled boundaries of the chip development process	183
5.3	The design pyramid [8]	184
5.4	Principle of boundary scan	186
5.5	<i>N</i> -Modular redundancy	187
5.6	Redundancy in (a) Space and (b) Time	191
5.7	CNN Stability in the presence of numerical errors	194
5.8	Autonomous pattern generation in a CNN	195
5.9	A Sunblade 1000 is Benchmarked (which has separate L1 instruction and data caches as well as a unified L2 cache)	197
5.10	MIPS code that performs better on Cache A	198
5.11	MIPS code that performs better on Cache B	198
5.12	Architectural approach for in-cloud file analysis service	200
5.13	The continuous coverage over time when a given number of engines are used in parallel [30]	200
6.1	DARPA information survivability research overview	206
6.2	The parameter space for novelty detection and isolation (NDI)	210
6.3	Set-point becomes non-optimal when abnormality occurs	210
6.4	V-chart for separated process and detection modeling, illustrating the dedicated but hierarchically layered focus over all control level	211
6.5	Classical FDI development phases	213
6.6	The high alarm limit is set too tight and is therefore sensitive to noise. The low alarm limit on the other hand is set too wide, and does not activate even though a plant trip occurs	215
6.7	The Control System receives Signals from the plant, where some signals are used for the controlling of the plant or displayed at the operator terminals. A subset of the signals is used for the alarm system. Whenever a signal exceeds its alarm limit, the alarm system activates. The applied signal processing methods consist of signal filtering, alarm	

	filtering, and alarm suppression, and do not interfere with the system's ability to control the process	216
6.8	Reactive and coordination layer in self-healing multi-agent smart grid control	217
6.9	Examples of SILENT CONTROL™ Signal Classes. Signals 1–6 are periodic, signals 7–11 are slowly varying, 12 and 13 are multiple steady-state signals and signal 14 contains outliers	219
6.10	CBAS algorithmic structure	222
6.11	Tradeoff between biometric methods	223
6.12	The block diagram of the pre-paid local traffic card	225
6.13	Block diagram of the manager payment module	227
6.14	A simple interaction between central control and nearest bus stops	228
6.15	Block diagram of conductor master controller module for long distance	228
6.16	Overview of the Interaction between the central control, the bus stops, the buses, and the cards	229
6.17	Data frame exchange between master and slave	230
6.18	A simple interaction between master and slave	232
7.1	The vision technology roadmap	247
7.2	Measuring method applied to the system model	249
7.3	Different cameras see different shadows	250
7.4	VASD development phase I	251
7.5	Thousand rays generation in the system model	251
7.6	Panoramic viewer concept and individual catadioptric panels	257
7.7	Image processing for multi-panel projection	259
7.8	Typical small office/home office (SoHo) structure [34]	263
7.9	Clouds on both sides of the internet!	274
8.1	The new clouds between sensory networks and server farms	281

List of Tables

3.1	Specification of two sigmoid transfer functions	96
4.1	RFID operating frequencies	124
4.2	Classification of tags	125
4.3	Different challenges faced by WSN and BSN (partly taken from Yang) [18]	133
4.4	Training and testing results for linear movement pattern	150
4.5	Training and testing results for border collisions	150
4.6	Training and testing results for circle wall collisions	151
6.1	Biometric Technologies (Adapted from RAND Report, June 2000)	220
7.1	Wireless home-network technologies	242
7.2	Some basic requirements for e-paper technology [15]	255
7.3	Panoramic viewer technology tradeoffs	256
7.4	Home-apps complexity estimates	272
7.5	Cloud apps for the home	273